

# หน่วยที่ 5

## Types

# เรื่องที่จะศึกษา

- Types, predefined and user-defined type
- The Stack and the Heap
- Value Types and Reference Types

# Type vs Programming language

C

Functions

Data types

C++

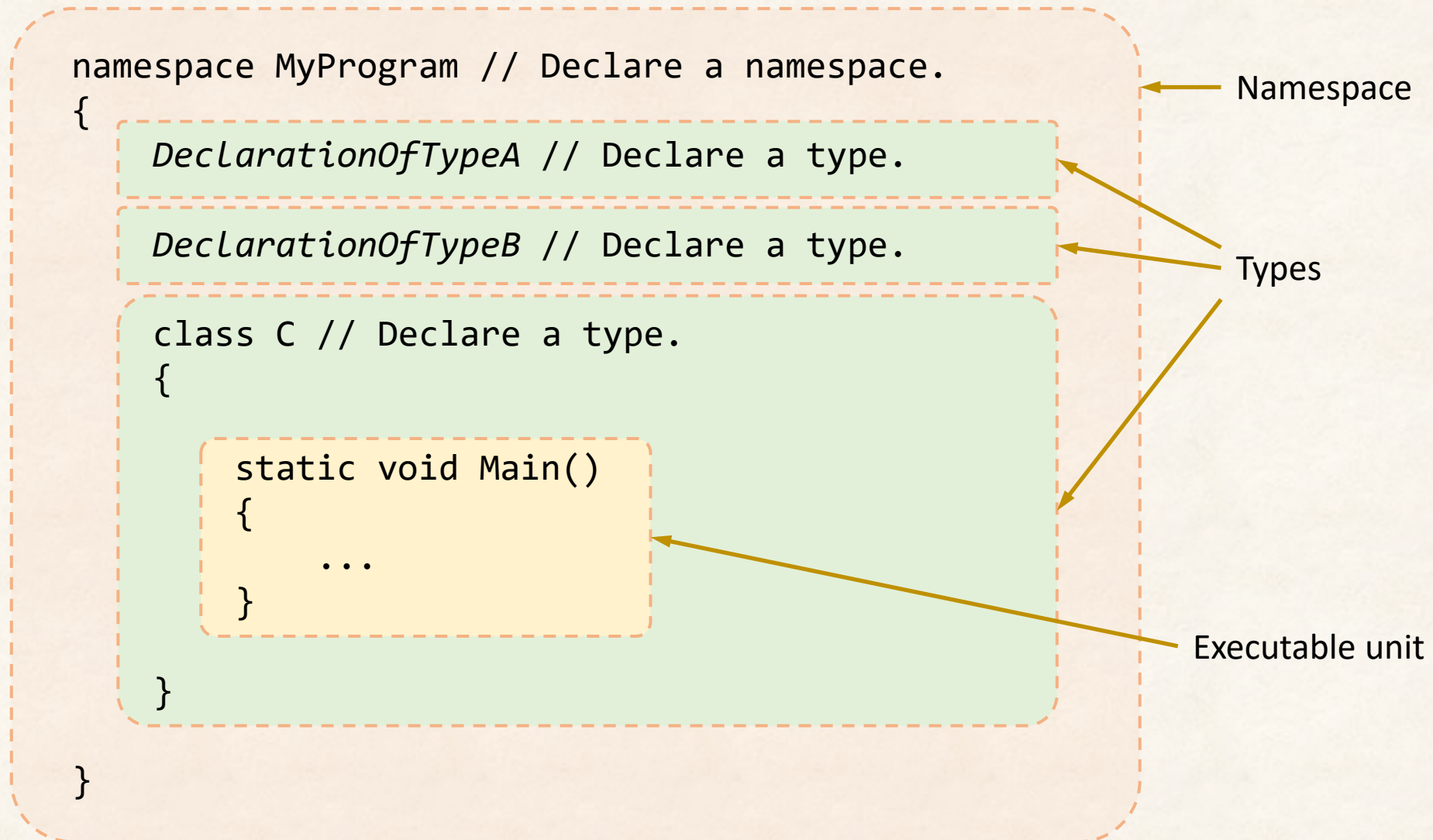
Functions

Classes

C#

Types

# Executable unit in c#

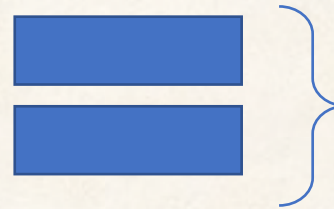
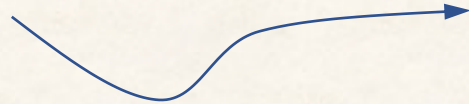


# A Type Is a Template

- Type คือ template สำหรับ Data structure (ไม่ใช่ตัว data structure)
- Type มีองค์ประกอบ 3 อย่าง
  - Name
  - Data structure
  - Behaviors & constraints

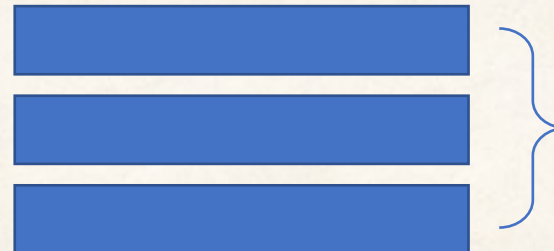
# A Type Is a Template

Name short	Structure 2 Bytes
Behavior 16-Bit Integer	



2 Instance  
of Type short

Name int	Structure 4 Bytes
Behavior 32-Bit Integer	



3 Instance  
of Type int

กระบวนการ instantiation ใช้ในการ  
สร้าง object หรือ instance จาก type

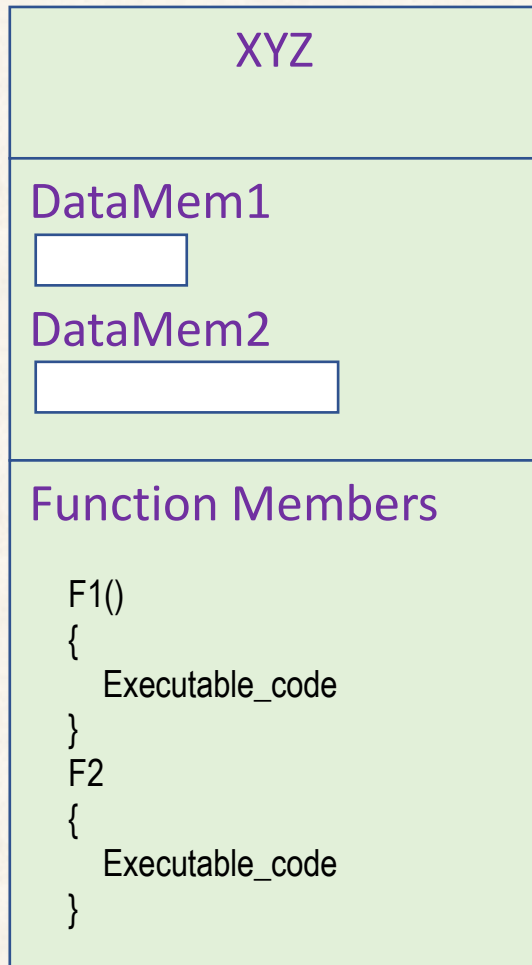
# องค์ประกอบของ Type

Name
Data Members
Function Members

- Type เก็บ data ได้เพียงหน่วยเดียวเรียกว่า Simple type
  - เช่น short, int, long
- Type สามารถเก็บ data ชนิดเดียวกันได้หลายหน่วย
  - เช่น Array และมี index ให้อ้างถึง data แต่ละหน่วย
- Type สามารถเก็บ data ได้หลากหลาย
  - เช่น Abstract Data Type



# องค์ประกอบของ Type





# Predefined Type

8-bit = 2 ชนิด

16-bit = 2 ชนิด

32-bit = 2 ชนิด

64-bit = 2 ชนิด

Non-numeric = 2 ชนิด

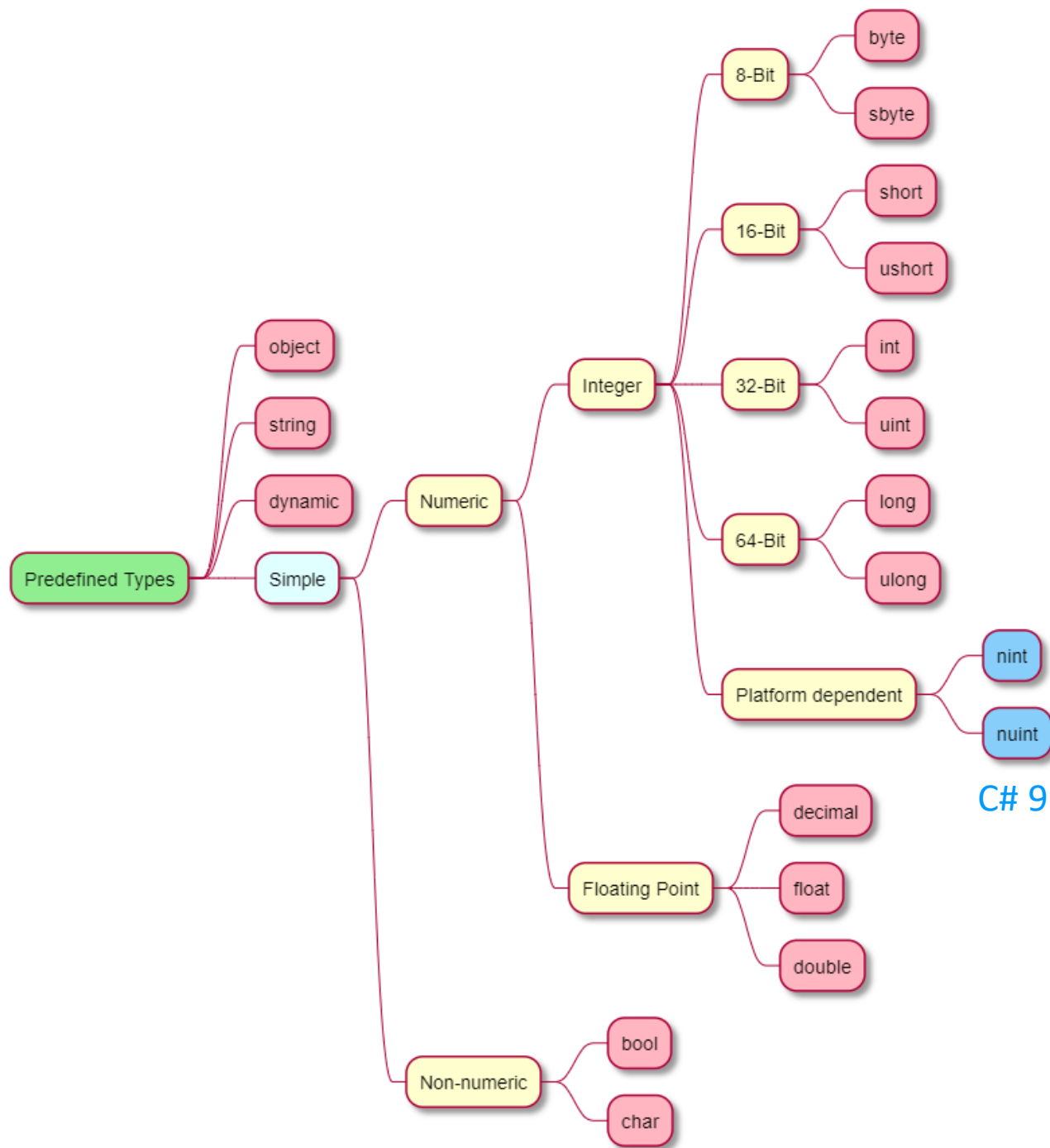
Floating point = 3 ชนิด

Object = 1 ชนิด

String = 1 ชนิด

Dynamic = 1 ชนิด

Native = 2 ชนิด



# Predefined (Simple Type)

Name	Meaning	Range	.NET Framework Type	Default Value
sbyte	8-bit signed integer	-128–127	System.SByte	0
byte	8-bit unsigned integer	0–255	System.Byte	0
short	16-bit signed integer	-32,768–32,767	System.Int16	0
ushort	16-bit unsigned integer	0–65,535	System.UInt16	0
int	32-bit signed integer	-2,147,483,648–2,147,483,647	System.Int32	0
uint	32-bit unsigned integer	0–4,294,967,295	System.UInt32	0

nint, nuint ขึ้นกับ platform จะมีขนาดเท่ากับ register ของ CPU

# Predefined (Simple Type)

Name	Meaning	Range	.NET Framework Type	Default Value
long	64-bit signed integer	-9,223,372,036,854,775,808–9,223,372,036,854,775,807	System.Int64	0
ulong	64-bit unsigned integer	0–18,446,744,073,709,551,615	System.UInt64	0
float	Single-precision float	$1.5 \times 10^{-45}$ – $3.4 \times 10^{38}$	System.Single	0.0f
double	Double-precision float	$5 \times 10^{-324}$ – $1.7 \times 10^{308}$	System.Double	0.0d
bool	Boolean	true, false	System.Boolean	false
char	Unicode character	U+0000–U+ffff	System.Char	\x0000
decimal	Decimal value with 28-significant-digit precision	$\pm 1.0 \times 10^{28}$ – $\pm 7.9 \times 10^{28}$	System.Decimal	0m

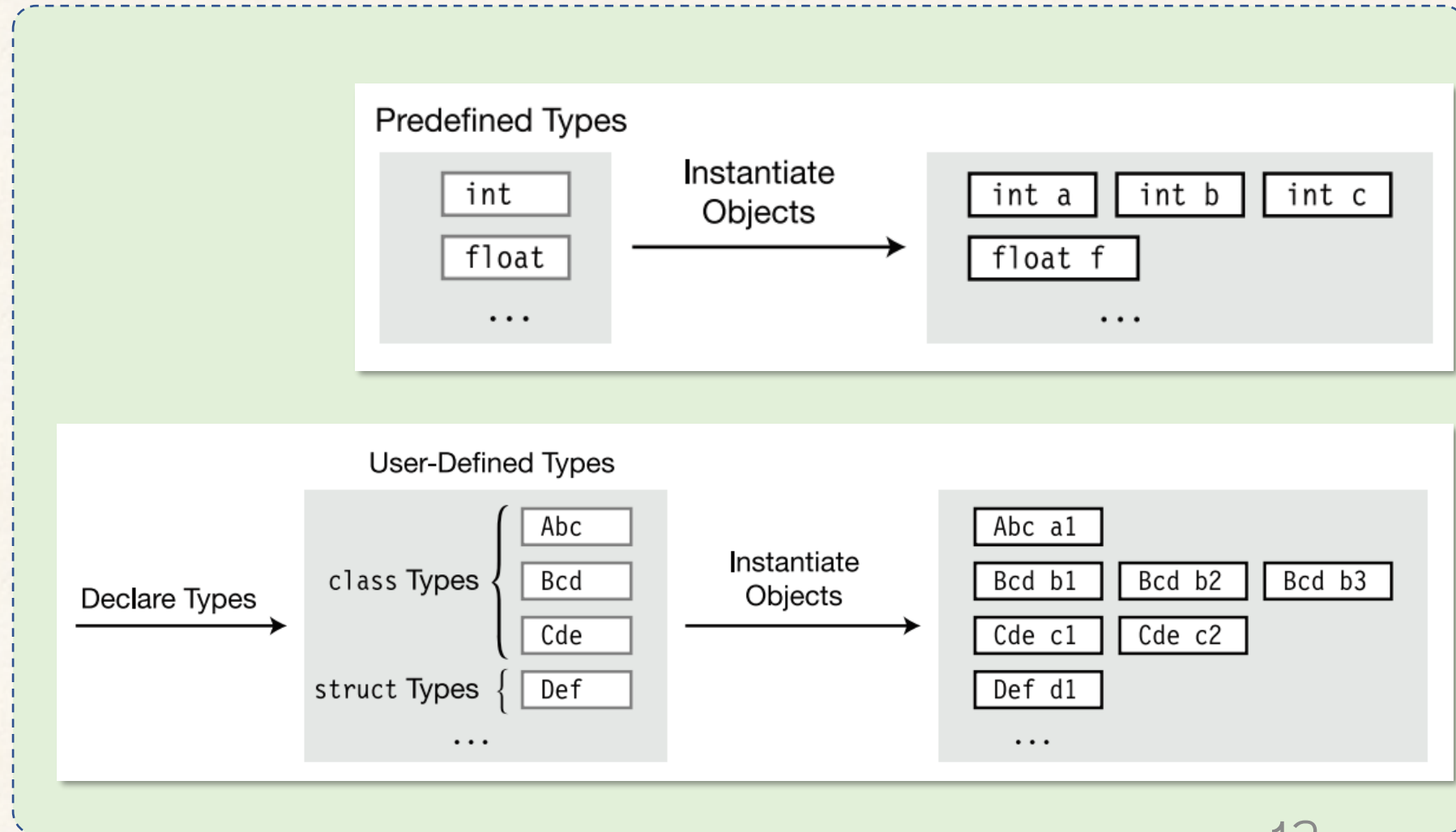
# Predefined (Nonsimple Type)

Name	Meaning	.NET Framework Type
object	The base class from which all other types, including the simple types are derived	System.Object
string	A sequence of zero or more Unicode characters	System.String
dynamic	A type designed to be used with assemblies written in dynamic languages	No corresponding .NET type

C# 9

# User-Defined Types

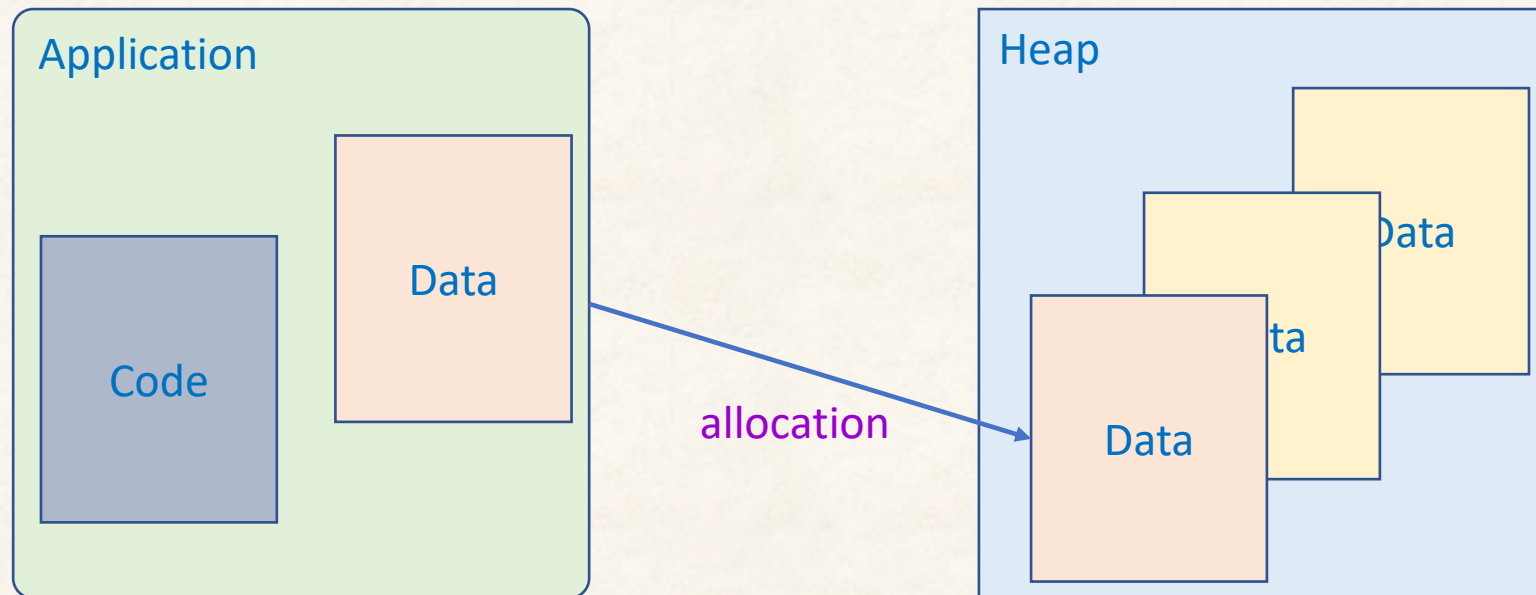
- class types
- struct types
- array types
- enum types
- delegate types
- interface types



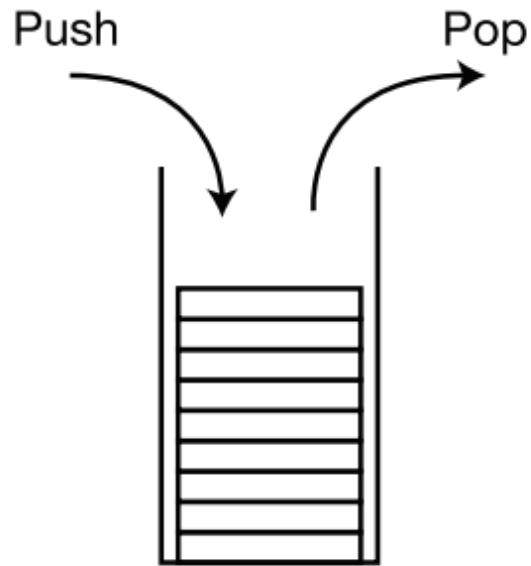


# The Stack and the Heap

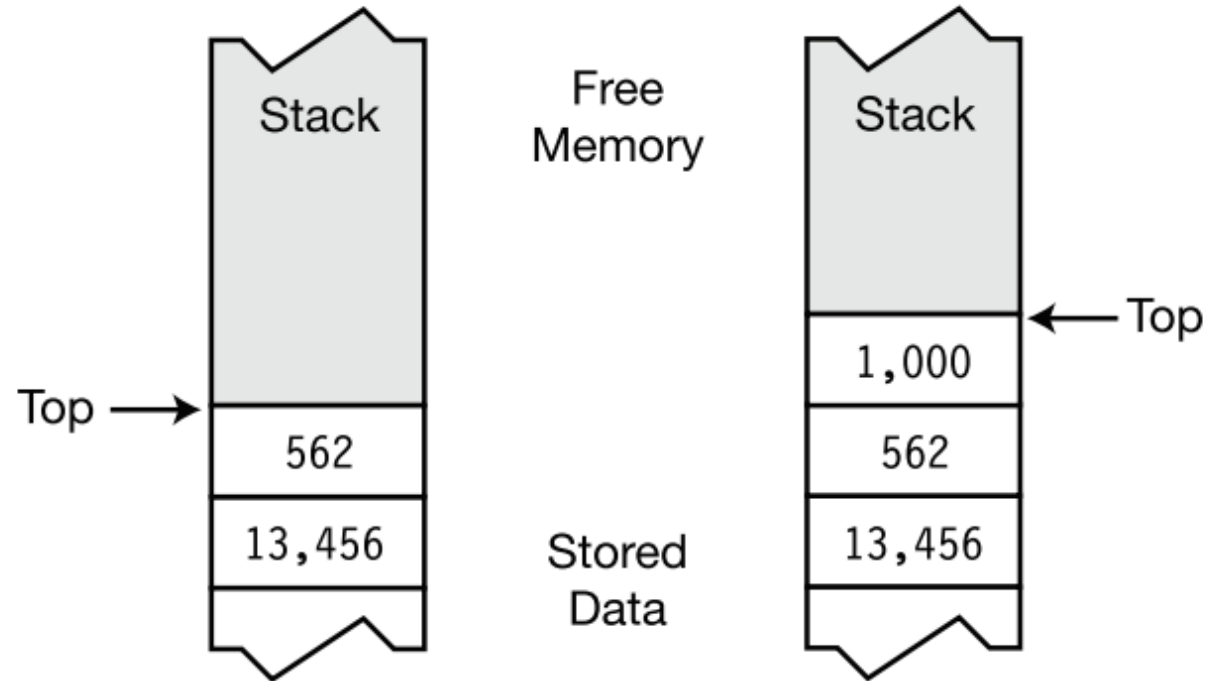
- ในขณะที่โปรแกรมทำงาน จะมีการจัดการกับ data ซึ่งต้องเก็บใน memory
  - Memory ที่เก็บ data มี 2 ส่วนคือ Stack และ Heap
  - Data จะถูกเก็บที่ไหนนั้น ขึ้นอยู่กับ type ของ data



# The Stack



Data items are *pushed* onto the top of the stack and *popped* from the top of the stack.



Pushing an integer (e.g., 1,000) onto the stack moves the top of the stack up.

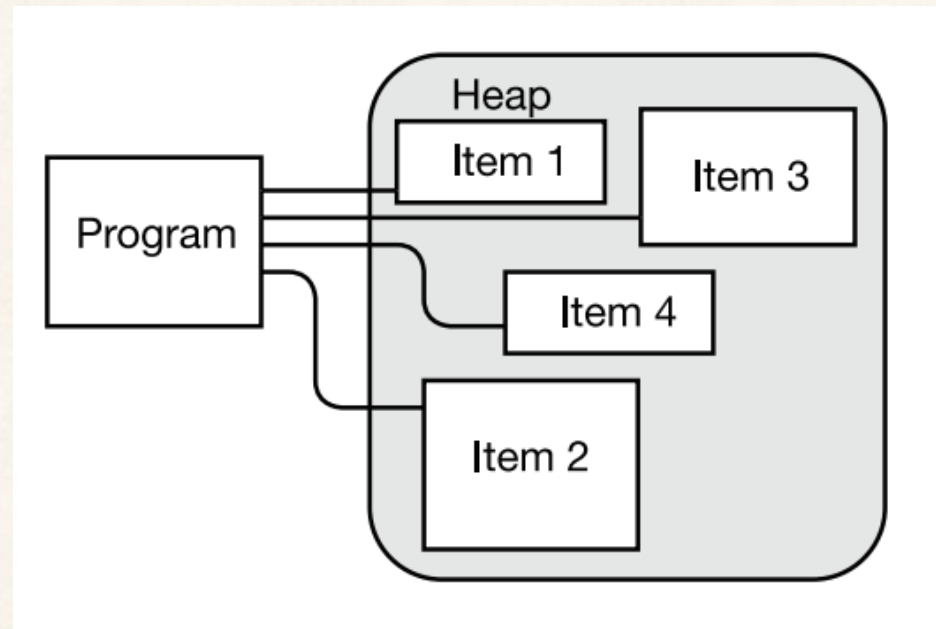


# The Stack

- Stack ของ Application มีหน้าที่
  - เก็บข้อมูล (ตัวแปรต่าง ๆ)
  - เก็บสถานะการทำงานของ Application
  - ส่งผ่านข้อมูล (parameters) ไปยัง method
- การนำข้อมูลเข้าไปเก็บใน stack เรียกว่าการ push
  - จะ push ที่ top of stack เสมอ (เสมือนการ insert ข้อมูล)
- การนำข้อมูลออกจาก stack เรียกว่าการ pop
  - จะ pop ที่ top of stack เสมอ (เสมือนการ delete ข้อมูล)

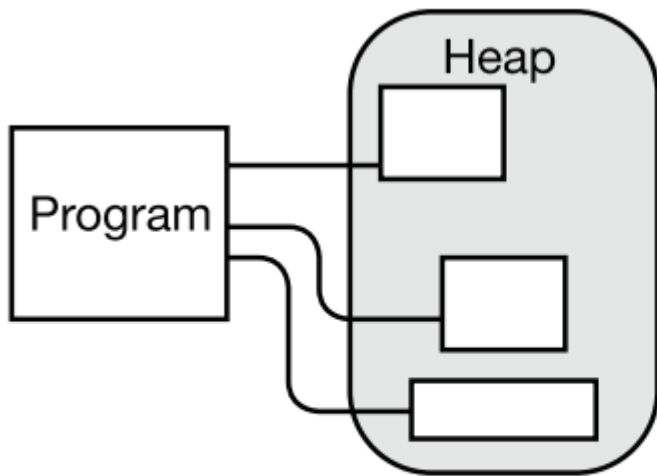
# The Heap

- Heap เป็นหน่วยความจำที่ถือครองโดยระบบ
- Application สามารถขอใช้หน่วยความจำ heap จากระบบ
  - การเข้าถึงข้อมูลจะต่างจาก stack ตรงที่สามารถเขียนอ่านได้ทุกตำแหน่งในทันที

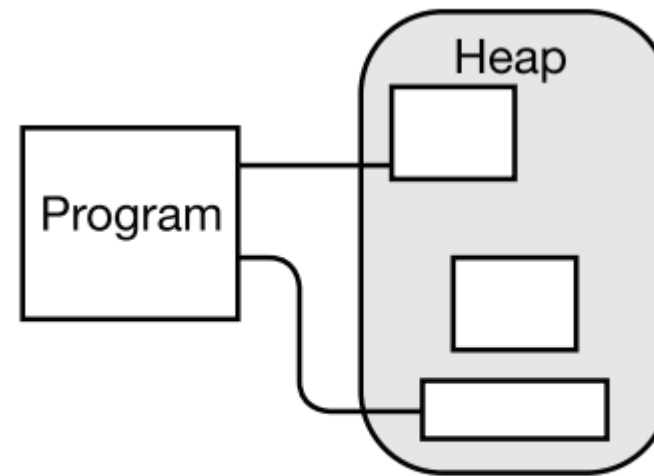


# The Heap

- เมื่อเลิกใช้ Application หรือเกิด Application crashed โดยที่ไม่มีการคืนหน่วยความจำที่ขอจากระบบ จะมี Garbage Collector (GC) นำหน่วยความจำนั้นไปคืนให้ระบบเมื่อถึงเวลาอันสมควร

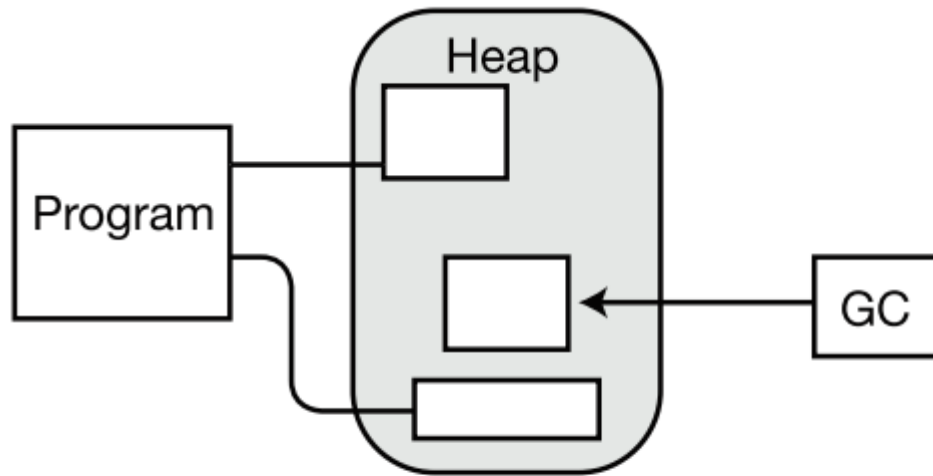


1. The program has stored three objects in the heap.

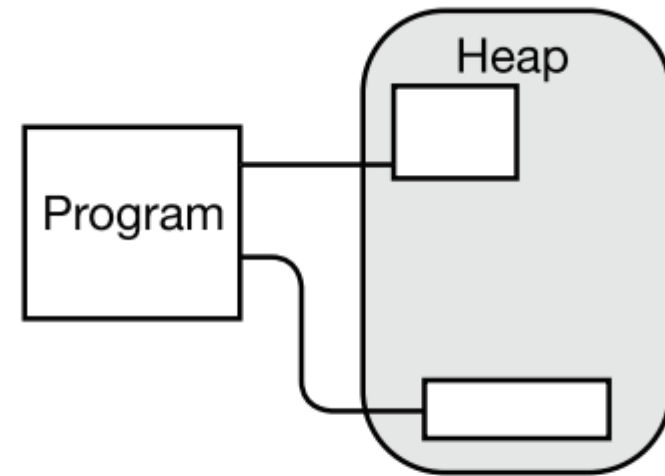


2. Later in the program, one of the objects is no longer used by the program.

# The Heap



3. The garbage collector finds the orphaned object and releases it.

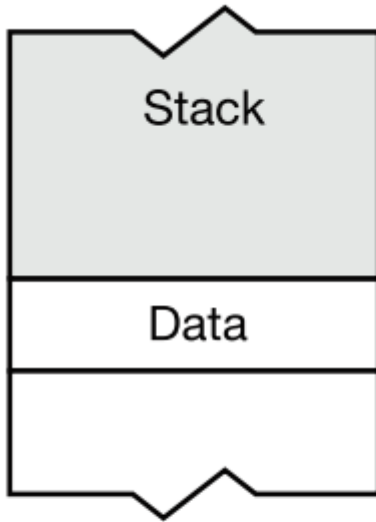


4. After garbage collection, the released object's memory is available for reuse.

# Value Types and Reference Types

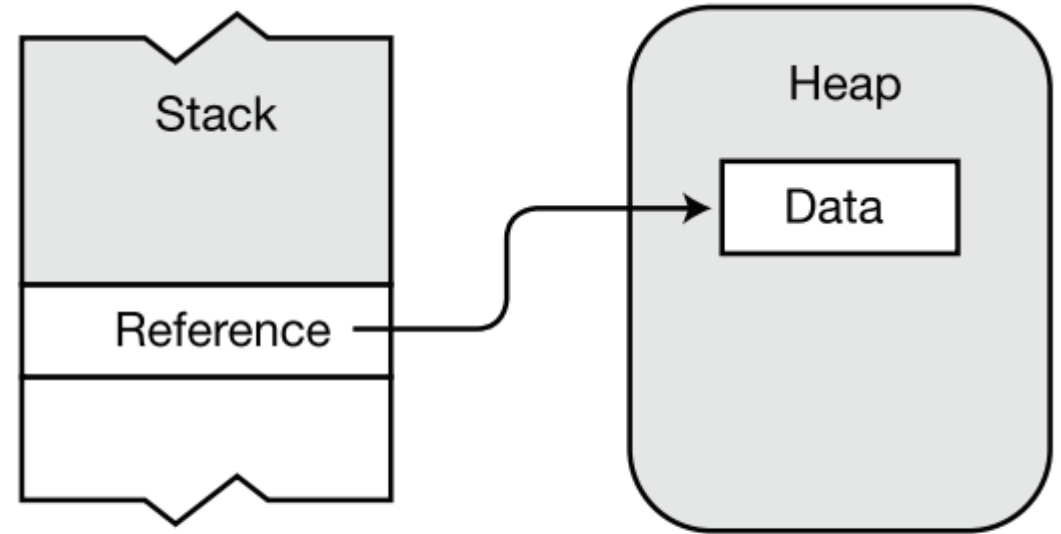
- ชนิดข้อมูลจะเป็นตัวกำหนดขนาดของหน่วยความจำที่ต้องการใช้จากระบบ
  - ทำให้รู้ว่าต้องเก็บไว้ที่ไหน (stack หรือ heap)
- เราสามารถแบ่งออกได้เป็น 2 ประเภท
  - Value type เปรียบเสมือนเหรียญกษาปณ์หรือธนบัตร ที่มีมูลค่าในตัว
  - Reference type เปรียบเสมือนบัตรเอทีเอ็ม หรือโฉนดที่ดิน

# Value Types and Reference Types



## Value Type Data

- The data is stored on the stack.



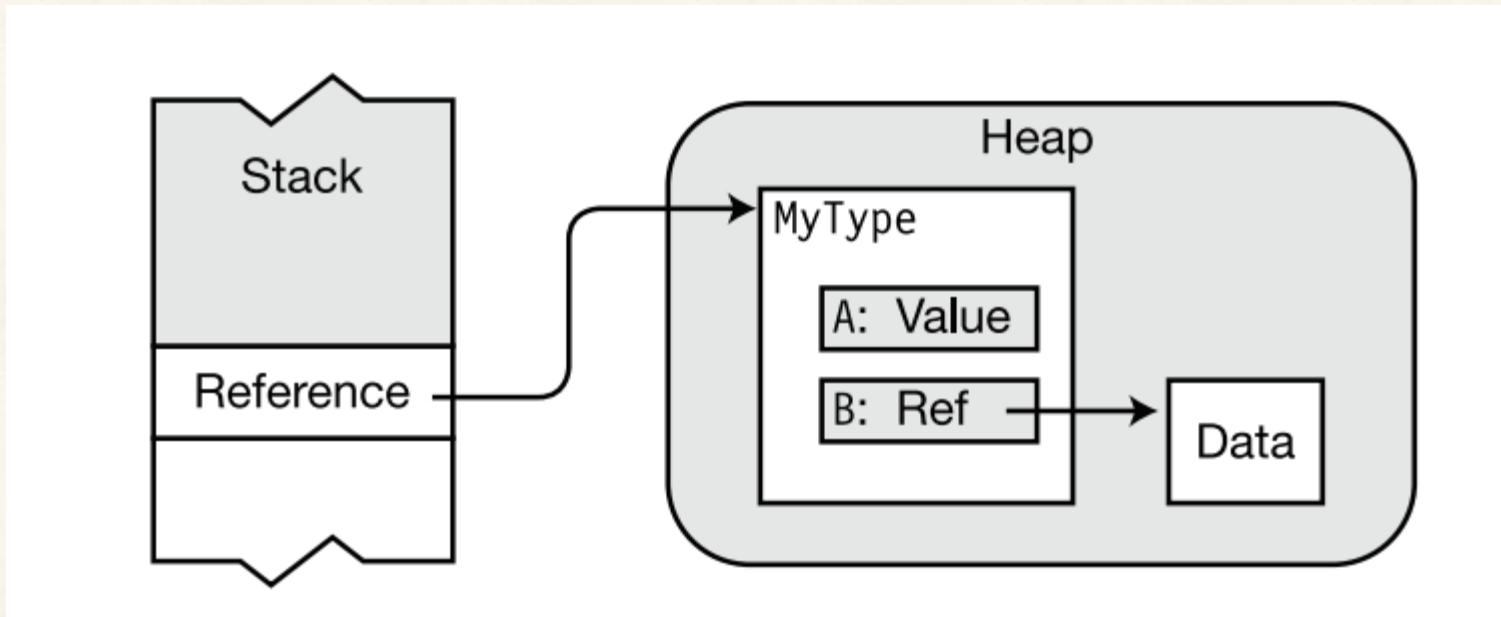
## Reference Type Data

- The data is stored in the heap.
- The reference is stored on the stack.



# Value Types สามารถเก็บบน heap ได้เช่นกัน

- ในกรณีที่เรสร้าง type ขึ้นจาก value type เมื่อสร้าง instance จะต้องสร้างไว้บน heap





# Value Types and Reference Types in C#

Value Types			Reference Types	
Predefined types	sbyte byte		float	object
	short	ushort	double	string
	int	uint	char	dynamic
	long	ulong	decimal	
	bool			
User-defined types	struct			class
	enum			interface
				delegate
				array